

Tanta University- Thrid Year-Dr. Hatem 2016

Modeling and Simulation Course

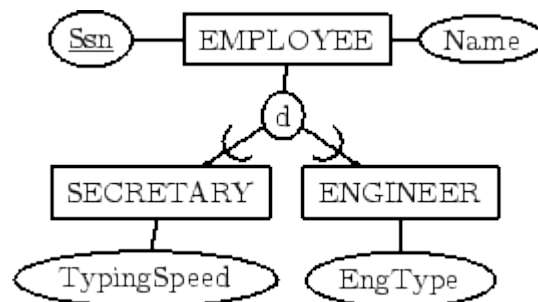
Enhanced ER Diagrams

Enhanced (Extended) ER Diagrams

- Contain all the basic modeling concepts of an ER Diagram
- Adds additional concepts:
 - Specialization/generalization
 - Subclass/super class
 - Categories
 - Attribute inheritance
- Extended ER diagrams use some object-oriented concepts such as inheritance.
- EER is used to model concepts more accurately than the ER diagram.

Sub classes and Super classes

- In some cases, and entity type has numerous sub-groupings of its entities that are meaningful, and need to be explicitly represented, because of their importance.
- For example, members of entity Employee can be grouped further into Secretary, Engineer, Manager, Technician, Salaried_Employee.
- The set listed is a subset of the entities that belong to the Employee entity, which means that every entity that belongs to one of the sub sets is also an Employee.
- Each of these sub-groupings is called a subclass, and the Employee entity is called the super-class.

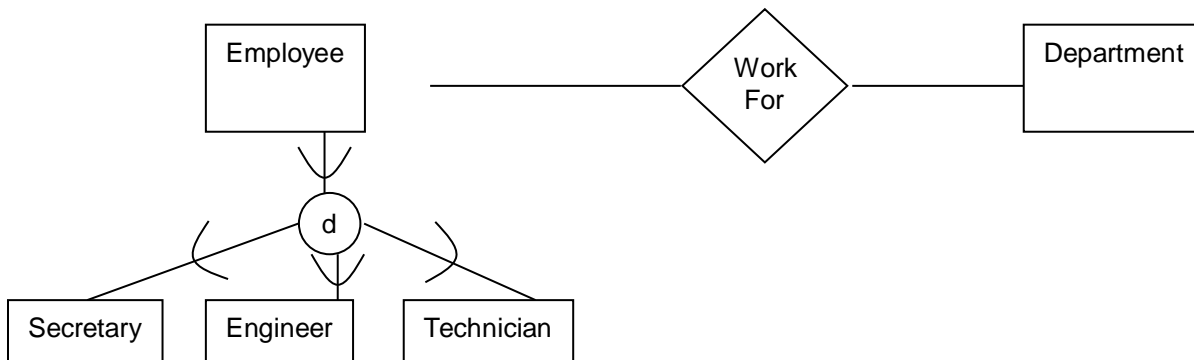


- An entity cannot only be a member of a subclass; it must also be a member of the super-class.

- An entity can be included as a member of a number of sub classes, for example, a Secretary may also be a salaried employee, however not every member of the super class must be a member of a sub class.

Type Inheritance

- The type of an entity is defined by the attributes it possesses, and the relationship types it participates in.
- Because an entity in a subclass represents the same entity from the super class, it should possess all the values for its attributes, as well as the attributes as a member of the super class.
- This means that an entity that is a member of a subclass inherits all the attributes of the entity as a member of the super class; as well, an entity inherits all the relationships in which the super class participates.

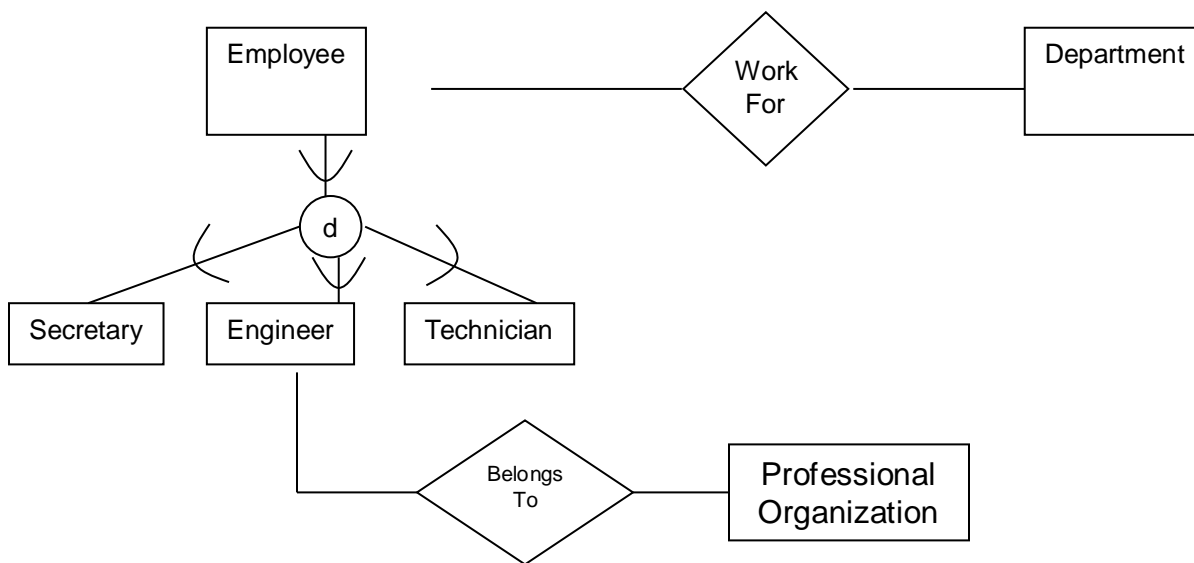


Specialization

- The process of defining a set of subclasses of a super class.
- Specialization is the top-down refinement into (super) classes and subclasses
- The set of sub classes is based on some distinguishing characteristic of the super class.
- For example, the set of sub classes for Employee, Secretary, Engineer, Technician, differentiates among employee based on job type.
- There may be several specializations of an entity type based on different distinguishing characteristics.
- Another example is the specialization, Salaried_Employee and Hourly_Employee, which distinguish employees based on their method of pay.

Notation for Specialization

- To represent a specialization, the subclasses that define a specialization are attached by lines to a circle that represents the specialization, and is connected to the super class.
- The subset symbol (half-circle) is shown on each line connecting a subclass to a super class, indicates the direction of the super class/subclass relationship.
- Attributes that only apply to the sub class are attached to the rectangle representing the subclass. They are called specific attributes.
- A sub class can also participate in specific relationship types. See Example.



Reasons for Specialization

- Certain attributes may apply to some but not all entities of a super class. A subclass is defined in order to group the entities to which the attributes apply.
- The second reason for using subclasses is that some relationship types may be participated in only by entities that are members of the subclass.

Summary of Specialization

Allows for:

- Defining set of subclasses of entity type
- Create additional specific attributes for each sub class
- Create additional specific relationship types between each sub class and other entity types or other subclasses.

Generalization

- The reverse of specialization is generalization.
- Several classes with common features are generalized into a super class.
- For example, the entity types Car and Truck share common attributes License_PlateNo, VehicleID and Price, therefore they can be generalized into the super class Vehicle.

Constraints on Specialization and Generalization

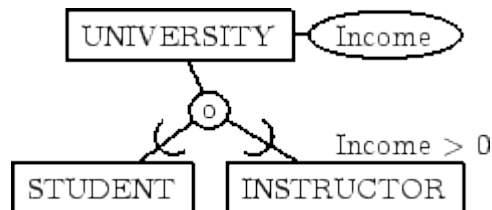
- Several specializations can be defined on an entity type.
- Entities may belong to subclasses in each of the specializations.
- The specialization may also consist of a single subclass, such as the manager specialization, in this case we don't use the circle notation.

Types of Specializations

Predicate-defined or Condition-defined specialization

- Occurs in cases where we can determine exactly the entities of each subclass by placing a condition of the value of an attribute in the super class.
- An example is where the Employee entity has an attribute, Job Type. We can specify the condition of membership in the Secretary subclass by the condition, JobType="Secretary"

Another Example:



- The condition is called the defining predicate of the sub class.
- The condition is a constraint specifying exactly those entities of the Employee entity type whose attribute value for Job Type is Secretary belong to the subclass.
- Predicate defined subclasses are displayed by writing the predicate condition next to the line that connects the subclass to the specialization circle.

Attribute-defined specialization

- If all subclasses in a specialization have their membership condition on the same attribute of the super class, the specialization is called an attribute-defined specialization, and the attribute is called the defining attribute.
- Attribute-defined specializations are displayed by placing the defining attribute name next to the arc from the circle to the super class.

User-defined specialization

- When we do not have a condition for determining membership in a subclass the subclass is called user-defined.
- Membership to a subclass is determined by the database users when they add an entity to the subclass.

Disjointness/Overlap Constraint

- Specifies that the subclass of the specialization must be disjoint, which means that an entity can be a member of, at most, one subclass of the specialization.
- The d in the specialization circle stands for disjoint.
- If the subclasses are not constrained to be disjoint, they overlap.
- Overlap means that an entity can be a member of more than one subclass of the specialization.
- Overlap constraint is shown by placing an o in the specialization circle.

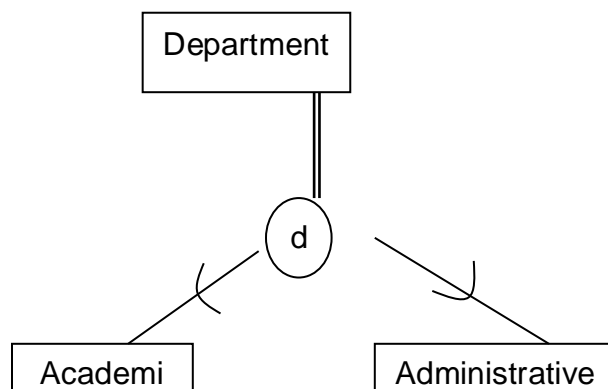
Completeness Constraint

- The completeness constraint may be either total or partial.
- A **total specialization** constraint specifies that every entity in the superclass must be a member of at least one subclass of the specialization.
- Total specialization is shown by using a double line to connect the super class to the circle.
- A single line is used to display a **partial specialization**, meaning that an entity does not have to belong to any of the subclasses.

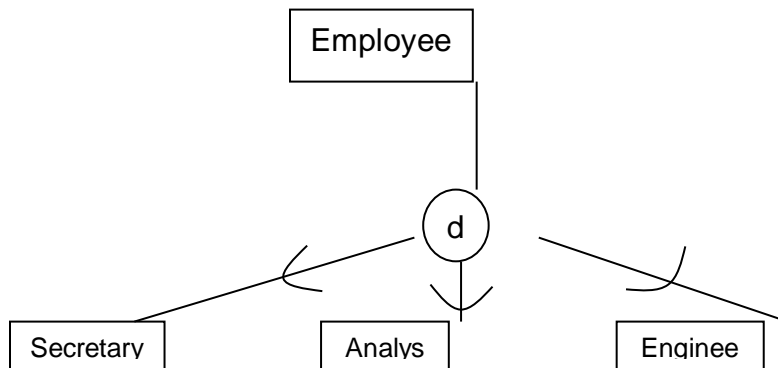
Disjointness vs. Completeness

- Disjoint constraints and completeness constraints are independent. The following possible constraints on specializations are possible:

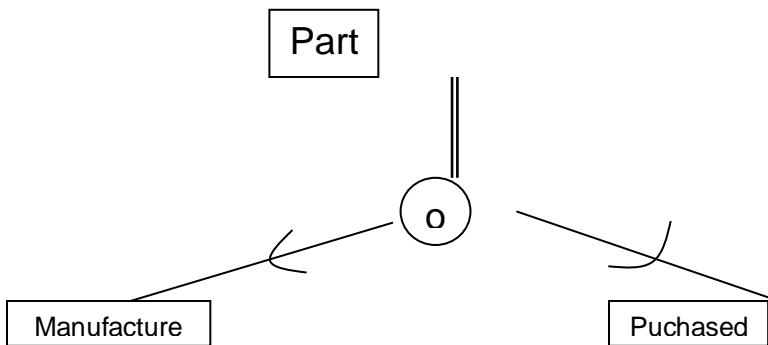
Disjoint, total



Disjoint, partial



Overlapping, total



Overlapping, partial

